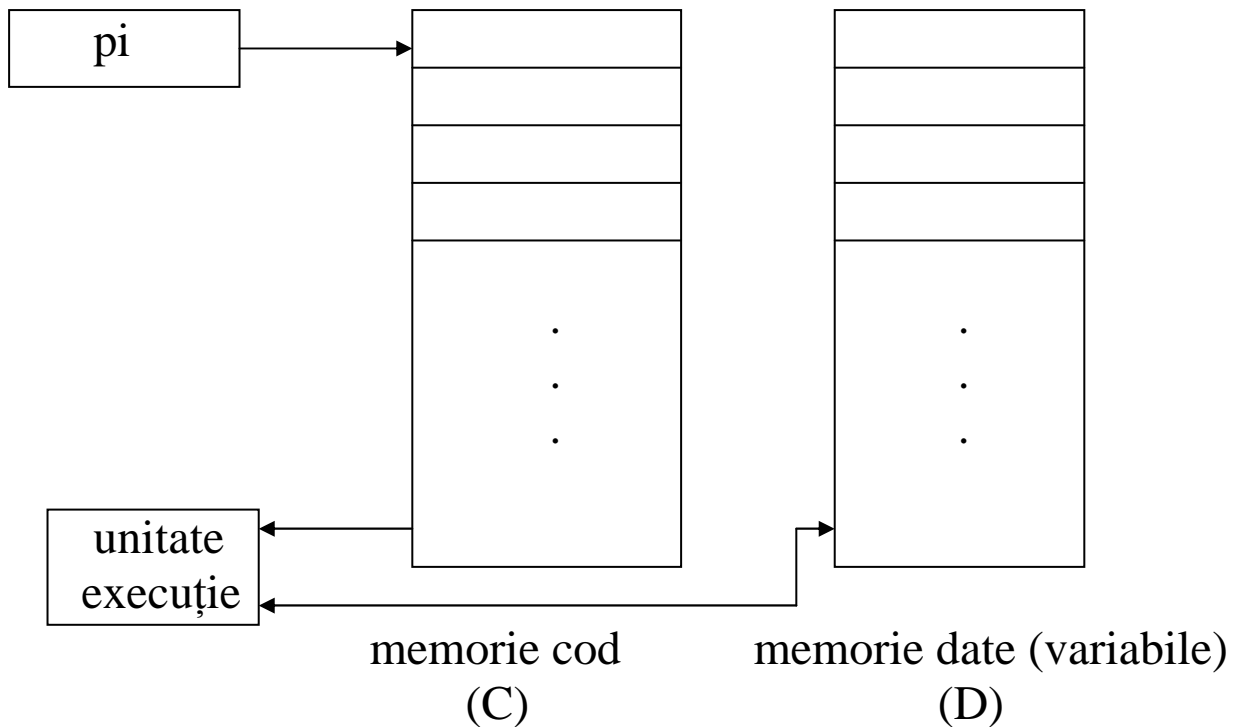


STRUCTURA UNUI PROCESOR VIRTUAL



SEMANTICA OPERAȚIONALĂ PENTRU for

```

for i := first to last do
    begin
        . . .
    end

```

s-ar defini ca :

```

    i := first
loop : if i > last goto out
    . . .
    i := i + 1
    goto loop
out : . . . .

```

SEMANTICA AXIOMATICĂ

Considerăm o instrucțiune S.

*Un **predicat P** care trebuie să fie adevărat după ce s-a executat S, se numește postcondiție pentru S.*

*Un **predicat Q** care fiind adevărat, S se execută normal și postcondiția P este adevărată, se numește precondiție pentru S și P.*

Notatie :

$$Q \{ S \} P$$

Exemplu :

$S : x := y + 1 \quad (\text{numere întregi})$

$P : x > 0 \qquad y = 3 \{ x := y + 1 \} x > 0$

$Q : y = 3$

$Q : y > -1 \qquad y > -1 \{ x := y + 1 \} x > 0$

O funcție semantică de forma :

$$axsem (S, P) = W$$

furnizează pentru o anumită construcție de limbaj S și postcondiție P, precondiția cea mai slabă corespunzătoare.

PENTRU INSTRUCȚIUNEA DE ATRIBUIRE

$$axsem (x := E, P) = Px \rightarrow E$$

unde $Px \rightarrow E$ reprezintă predicatul P , în care toate aparițiile lui x s-au înlocuit cu E .

$$Px \rightarrow E \{ x := E \} P$$

La exemplul de mai sus

$$y > -1 \{ x := y + 1 \} x > 0$$

vrem să folosim funcția semantică ***axsem*** pentru a afla dacă instrucțiunea

$$x := x + 3 \text{ produce un rezultat } x > 8$$

ce răspuns ne dă funcția semantică ?

$$axsem (x := x + 3, x > 8) = x > 5$$

prin urmare, dacă $x > 5$ înainte de atribuire, atunci $x > 8$ după aceasta.

PENTRU O SECVENȚĂ DE INSTRUCȚIUNI

Dacă

$$axsem (S1, P) = Q$$

$$axsem (S2, Q) = R$$

atunci, corespunzător secvenței de instrucțiuni $S2 ; S1$ avem

$$axsem (S2; S1, P) = R$$

PENTRU INSTRUCȚIUNEA if

if B then L1 else L2 end if

B : condiție

L1, L2 : secvențe de instrucțiuni

axsem (instr-if, P) = B \sqsubset *axsem* (L1, P) and not B \sqsubset *axsem* (L2, P)

Exemplu :

considerăm

$$\text{if } \overbrace{x \geq y}^{\mathbf{B}} \text{ then } \overbrace{\text{max} := x}^{\mathbf{L1}} \text{ else } \overbrace{\text{max} := y}^{\mathbf{L2}} \text{ end if}$$

Dacă secvența determină în mod corect maximul dintre x și y , atunci postcondiția de mai jos trebuie să fie adevărată, indiferent de situația inițială.

($x \geq y$ and $\text{max} = x$) or ($y > x$ and $\text{max} = y$)

Considerând această postcondiție P, să vedem care e preconditionia corespunzătoare :

$(x \geq y) \sqsubset ((x \geq y \text{ and } x = x) \text{ or } (y > x \text{ and } x = y)) \text{ and }$
 $\text{not } (x \geq y) \sqsubset ((x \geq y \text{ and } y = x) \text{ or } (y > x \text{ and } y = y)) = \text{true}$

Prin urmare, rezultatul este corect în mod necondiționat.

SEMANTICA DENOTAȚIONALĂ

Starea sistemului va fi reprezentată formal printr-un triplet :

$$S = \langle \text{mem}, i, o \rangle$$

unde :

- **mem** este o funcție care reprezintă starea memoriei ;
Id fiind mulțimea tuturor identificatorilor posibili

$$\text{mem} : Id \longrightarrow Z \cup \{ \text{nedef} \}$$

(**nedef** e valoarea unui identificator nedefinit);

Z : mulțimea valorilor întregi

- *i, o* : reprezintă secvențele de intrare și de ieșire ;
valorile lor pot fi secvențe de numere întregi, eventual
secvența vidă.

PENTRU O EXPRESIE ARITMETICĂ

Semantica evaluării unei expresii se va descrie printr-o funcție :

$$dsemEx : EX \times S \longrightarrow Z \cup \{ \text{eroare} \}$$

S : mulțimea stărilor

EX : mulțimea expresiilor

unde :

$$dsemEx (E, s) = \text{eroare}$$

dacă $s = \langle \text{mem}, i, o \rangle$ și $\text{mem}(v) = \text{nedef}$,
pentru o variabilă v din E ; altfel

$$dsemEx (E, s) = e$$

dacă $s = \langle \text{mem}, i, o \rangle$ și e este rezultatul
evaluării expresiei E după înlocuirea fiecărui
identificator v din E - cu valoarea $\text{mem}(v)$.

PENTRU INSTRUCȚIUNEA DE ATRIBUIRE

Semantica atribuirii presupune și o transformare a stării :

$$dsemAt : AT \times S \longrightarrow S \cup \{ \text{eroare} \}$$

AT : mulțimea instrucțiunilor de atribuire

unde:

$$dsemAt(x := E, s) = \text{eroare}$$

dacă $dsemEx(E, s) = \text{eroare}$; altfel

$$dsemAt(x := E, s) = s'$$

unde $s = \langle \text{mem}, i, o \rangle$, $s' = \langle \text{mem}', i', o' \rangle$

$i' = i$, $o' = o$,

$\text{mem}'(y) = \text{mem}(y)$, pentru $\forall y \neq x$

$\text{mem}'(x) = dsemEx(E, s)$

PENTRU INSTRUCȚIUNEA DE CITIRE

Citirea, pe care o notăm $x \leftarrow read$, va afecta starea sistemului atât din punct de vedere al memoriei, cât și al șirului de intrare.

$$dsemCt : CT \times S \longrightarrow S \cup \{\text{eroare}\}$$

CT : mulțimea instrucțiunilor de citire

unde :

$$dsemCt(x \leftarrow read, s) = \text{eroare}$$

dacă $s = \langle \text{mem}, i, o \rangle$ și i este vid ; altfel

$$dsemCt(x \leftarrow read, s) = s'$$

unde $s = \langle \text{mem}, i, o \rangle$; $s' = \langle \text{mem}', i', o' \rangle$

$o = o'$, $i = i'$

$\text{mem}'(y) = \text{mem}(y)$, pentru $\forall y \neq x$

$\text{mem}'(x) = I$

PENTRU O SECVENȚĂ DE INSTRUCȚIUNI

$$dsemSi : SI \times S \longrightarrow S \cup \{\text{eroare}\}$$

SI : mulțimea secvențelor de instrucțiuni

unde :

- cazul listei vide, ϵ

$$dsemSi (\epsilon, s) = s$$

- cazul listei formate din instrucțiunea T urmată de lista L : $T ; L$

$$dsemSi (T ; L, s) = \text{eroare}$$

dacă $dsem(T, s) = \text{eroare}$; altfel

$$dsemSi (T ; L, s) = dsemSi (L, dsem (T, s))$$

considerând că $dsem$ descrie semantica lui T .

PENTRU INSTRUCȚIUNEA if

if B then L1 else L2 end if

B : expresie cu valoare $\begin{cases} = 0 & (\text{fals}) \\ \neq 0 & (\text{adevărat}) \end{cases}$

L1, L2 : secvențe de instrucțiuni

$dsemIf : IF \times S \longrightarrow S \cup \{\text{eroare}\}$

IF : mulțimea instrucțiunilor if

unde :

$dsemIf (\text{if } B \text{ then } L1 \text{ else } L2 \text{ end if}, s) = \text{eroare}$
dacă $dsemEx (B, s) = \text{eroare}$; altfel

$dsemIf (\text{if } B \text{ then } L1 \text{ else } L2 \text{ end if}, s)$
 $= dsemSi (L1, s) \text{ dacă } dsemEx (B, s) \neq 0$; altfel
 $= dsemSi (L2, s)$

PENTRU INSTRUCȚIUNEA LOOP

while B do L end while

B : expresie cu valoare $\begin{cases} = 0 \text{ (fals)} \\ \neq 0 \text{ (adevărat)} \end{cases}$

L : secvență de instrucțiuni

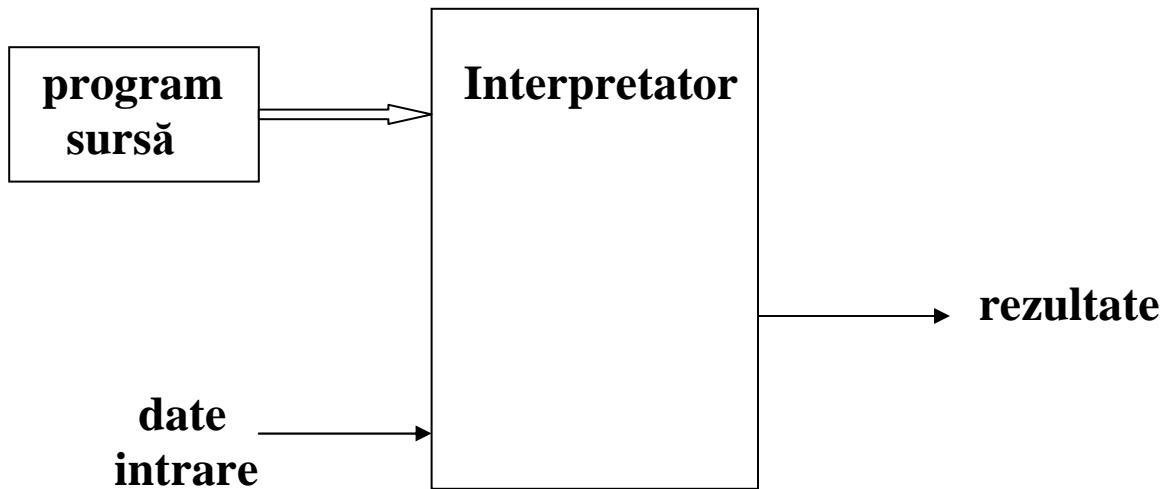
*dsem*While : WHILE x S \longrightarrow S \cup { eroare }

WHILE : mulțimea instrucțiunilor **while**

*dsem*While (while B do L end while, s) =

$$\left\{ \begin{array}{l} = \textbf{eroare} \text{ dacă } dsemEx(B, s) = \text{eroare} ; \text{ altfel} \\ = s \text{ dacă } dsemEx(B, s) = 0 ; \text{ altfel} \\ = \textbf{eroare} \text{ dacă } dsemSi(L, s) = \text{eroare} ; \text{ altfel} \\ = dsemWhile(\text{while } B \text{ do } L \text{ end while}, dsemSi(L, s)) \end{array} \right.$$

INTERPRETAREA



TRANSLATAREA

